
Family Map Client Specification

Last Modified: January 3, 2018

Contents

Contents	2
Acknowledgements	4
Introduction	4
Purposes	4
Family Map Client: A Quick Overview	4
Activities.....	5
Main Activity	5
Login Fragment	6
Layout.....	6
Functionality.....	6
Map Fragment	7
Layout.....	7
Functionality.....	7
Person Activity.....	8
Layout.....	8
Functionality.....	8
Event Activity.....	9
Layout.....	9
Functionality.....	9
Settings Activity	10
Logout.....	10
Re-Sync	10
Map Type	11
Enable/Disable Relationship Lines	11
Up Button / Options Menu	12
Filter Activity	12
Filter Types	12
Up Button / Options Menu	13
Search Activity.....	14
Layout.....	14
Resources.....	15
Family Map Server	15

Map Libraries for Android	15
Drawing Icons.....	16
JSON Parsing.....	17
Automated Tests.....	18
Source Code/Test Case Evaluation	18
Grading.....	18

Acknowledgements

The Family Map project was created by Jordan Wild. Thanks to Jordan for this significant contribution.

Introduction

Family Map is an Android application that provides a geographical view into your family history. One of the most exciting aspects of researching family history is discovering your origins. Family Map provides a detailed view into where you came from.

Purposes

The purposes of this project are to dig a little deeper into the following:

- Object-Oriented Design
- User Interface Programming
- Native Android Development
- Calling external services through Web APIs
- Unit Testing

Family Map Client: A Quick Overview

The Family Map Android application consists of six main views:

- Main Activity (Login and Top-level map)
- Event Activity (Lower-level maps)
- Person Activity
- Settings Activity
- Filter Activity
- Search Activity

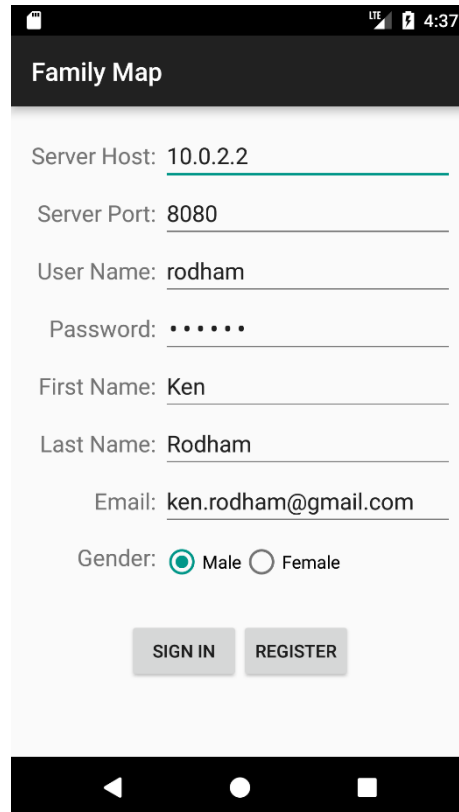
The Family Map Client application uses two external services:

- Family Map Server - Used for user management, data generation, and requesting data (this is the Family Map Server you created in the previous project)
- Google Maps v2 for Android or Amazon Maps v2 for Kindle - Used for displaying maps

Activities

Main Activity

When the application runs, it displays the Main Activity. Before the user logs in, the Main Activity displays a Login Fragment, as shown below. The Login Fragment allows the user to login if they already have a user account, or to register to create a new user account.

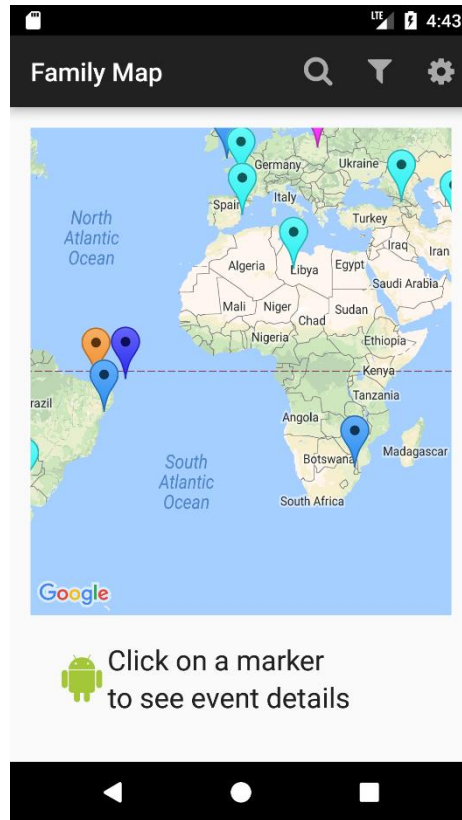


The screenshot shows a mobile application interface titled "Family Map". It contains a login form with the following fields and controls:

- Server Host: 10.0.2.2
- Server Port: 8080
- User Name: rodham
- Password: •••••
- First Name: Ken
- Last Name: Rodham
- Email: ken.rodham@gmail.com
- Gender: ☒ Male ☐ Female
- Buttons: SIGN IN, REGISTER

The status bar at the top shows LTE, battery, and time 4:37. The bottom navigation bar shows standard Android navigation icons.

After successful sign-in or registration, the user is logged in, and the Main Activity switches from the Login Fragment to the Map Fragment, which displays markers for the events in the user's family history, as shown below.



Login Fragment

Layout

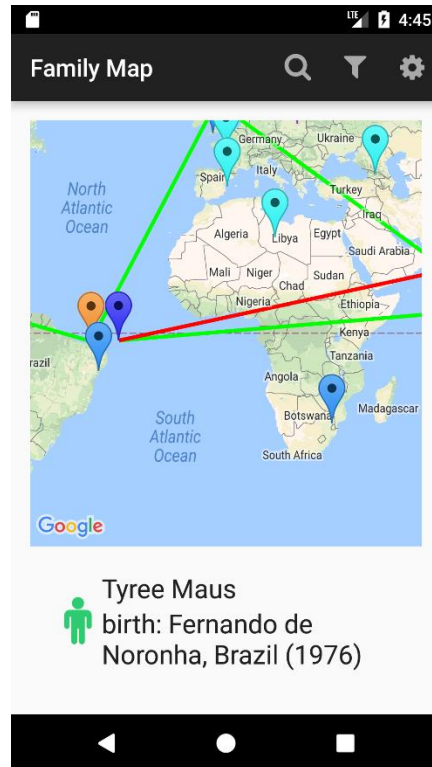
On the Login Fragment's layout, there are editable text fields where users can enter their server hostname / IP address, server port number, user name, password, first name, last name, and email address. There are also two radio buttons that allow users to specify their gender. Additionally, there are two buttons that allow the user to initiate sign-in with an existing user account or creation of a new user account. The Login Fragment does not display an up button or an options menu.

Functionality

In order to log in or register, the user must specify the server host and server port, which allows the app to contact the Family Map Server. To log in to an existing user account, the user must specify user name and password values, and click the "SIGN IN" button. The "SIGN IN" button is disabled unless the server host, server port, user name, and password values are filled in. To create a new user account and then log in to the new account, the user must specify values for all fields, and then click the "REGISTER" button. The "REGISTER" button is disabled unless all field values are filled in. If the sign-in / registration operation succeeds, the Login Fragment synchronizes the application with data from the Family Map Server. Synchronization pulls the data from the Family Map Server and stores it so that the application can access it later. If sign-in / registration or data synchronization fails, the Login Fragment displays an error message (i.e., Android "toast"), and allows the user to retry the operation. If sign-in / registration and synchronization succeed, the Main Activity displays a Map Fragment (described next).

Map Fragment

The Map Fragment takes data from the Family Map Server and displays it on an interactive map. Data is displayed according to the application's current settings and filters (see the Settings and Filter activities described later for more details).



Layout

On the Map Fragment's layout, there is an interactive map, and a place for event information to be displayed. When displayed in the Main Activity, the Map Fragment does not display an up button, but it does have an options menu that contains Search, Filter, and Settings icons/options.

Functionality

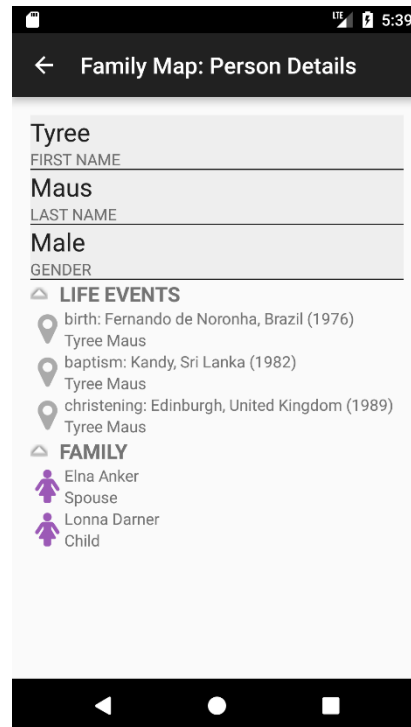
The options menu items allow the user to navigate to the Search Activity, Filter Activity, and Settings Activity, respectively. The interactive map allows the user to zoom in and out, select events from their family history, and view information about those events. Events are displayed on the map by markers.

The markers are color coordinated based on event type. The markers are clickable, updating the event information display when selected. The map also displays lines that represent relationships between events. The Map Fragment draws the line types that are enabled in the Settings Activity (i.e., life story, spouse, and family tree lines). The Map Fragment filters events based on filters defined and enabled in the Filter Activity. See the sections on the Settings and Filter activities for more details.

The event information display shows information about the selected event including the person's gender, the person's name, the event description, and event date. When the event information display is clicked, the Person Activity is created to display information about the person associated with the event (described next).

Person Activity

The Person Activity displays information about a person, including first name, last name, gender, and information about the person's family members and life events.



Layout

On the Person Activity's layout, there is a line for every descriptive field about a Person. There is also a collapsible list that displays the person's parents, spouse, and children. There is another collapsible list that displays a list of all events (that aren't being filtered) in the person's life story, listed in chronological order. There is an up button and no options menu.

A person's life events are ordered chronologically as follows:

1. Birth events, if present, are always first (whether they have a year or not)
2. Events with years, sorted primarily by year, and secondarily by description normalized to lower-case
3. Events without years sorted by description normalized to lower-case
4. Death events, if present, are always last (whether they have a year or not)

Functionality

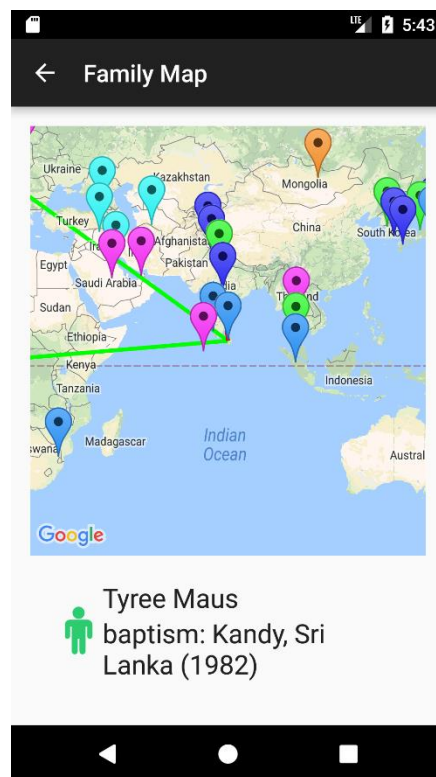
Clicking on a person in the relationship list triggers the creation of a new Person Activity with the information of the selected person. Clicking on an event triggers the creation of the

Event Activity, with the map zoomed in and centered on the location of the clicked-on event (described next).

Clicking the up button returns the user back to the top-level Main Activity. (Note, it does not open a new Main Activity, but rather returns to the original Main Activity by closing the current activity and all other intermediate activities that led to it.)

Event Activity

When the user clicks on an event in the Person or Search activities, an Event Activity is created to display information about the selected event. The Event Activity uses the same Map Fragment as the Main Activity to display an interactive map of events. However, when an Event Activity is created, the event clicked on by the user in the Person or Search activity is automatically selected and centered on the map, as shown below.



Layout

The Event Activity layout contains a Map Fragment that displays an interactive map and a display that shows information about the currently-selected event. When displayed in the Event Activity, the Map Fragment displays an up button and no options menu.

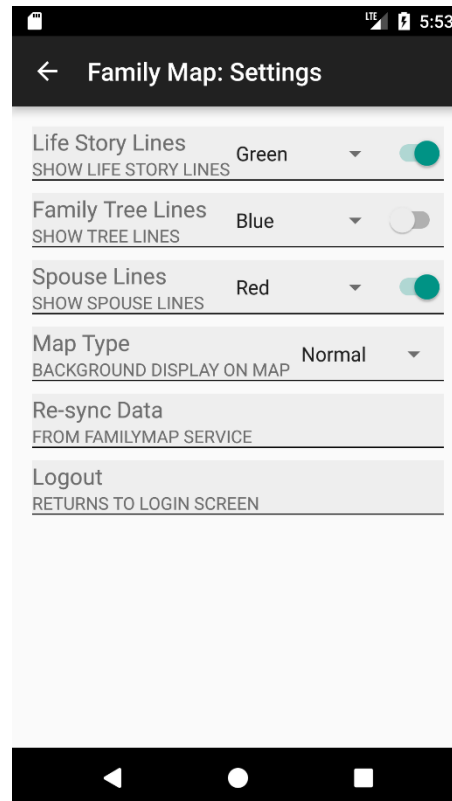
Functionality

The functionality of Event Activity is the same as that of Map Fragment described previously, with the added requirement that the event being viewed is automatically selected and centered on the map.

Clicking the up button returns the user back to the top-level Main Activity. (Note, it does not open a new Main Activity, but rather returns to the original Main Activity by closing the current activity and all other intermediate activities that led to it.)

Settings Activity

The Settings Activity allows the user to manage their account and modify certain aspects of the map's appearance. On the Settings Activity, there is a line for every actionable setting, as shown below.



Logout

Logging out destroys the user's session and returns them to the Main Activity. When it is re-displayed, the Main Activity presents the Login Fragment so the user can re-login if they desire.

Re-Sync

Re-sync with the Family Map Server. This destroys any locally cached family data and re-loads all required data from the Family Map Server. This is done without changing any of the application settings.

If the re-sync fails, an error message (i.e., Android "toast") is displayed, and the Settings Activity remains open, thus allowing the user to try again.

If the re-sync succeeds, the user is returned to the Main Activity. When it is re-displayed, the Main Activity presents the Map Fragment populated with the new data and no event selected.

Map Type

Change the map type to one of the following:

- Normal (Default), Hybrid, Satellite, or Terrain

Enable/Disable Relationship Lines

Enable and disable the types of relationship lines to be drawn on the map. Each line has a color chosen by the user. By default, each line should have a different color. The types of relationship lines are defined as follows:

Spouse Lines

A line is drawn linking the selected event to the birth event of the person's spouse (i.e., the person associated with the selected event). If there is no birth event recorded for the spouse, or if their birth event is currently invisible due to the current event filter settings, the earliest available event for the spouse is used instead. If the person has no recorded spouse, or the recorded spouse has no events, no line is drawn.

Family Tree Lines

Lines linking the selected event to the person's ancestors (i.e., the person associated with the selected event) are drawn as follows:

- A line is drawn between the selected event and the birth event of the selected person's father. If the person's father does not have a birth event, or if their birth event is currently invisible due to the current event filter settings, the earliest available event for the father is used instead. If the person has no recorded father, or the recorded father has no events, no line is drawn.
- A line is drawn between the selected event and the birth event of the selected person's mother. The same logic that applies to the father also applies to the mother.
- Lines are drawn recursively from parents' birth events to grandparents' birth events, from grandparents' birth events to great grandparents' birth events, etc. including all available generations. In all cases, if a person's birth event does not exist, or is invisible due to the current event filter settings, the earliest event in the person's life should be used instead of their birth event. As lines are drawn recursively up the family tree, they should become progressively and noticeably thinner.

Life Story Line

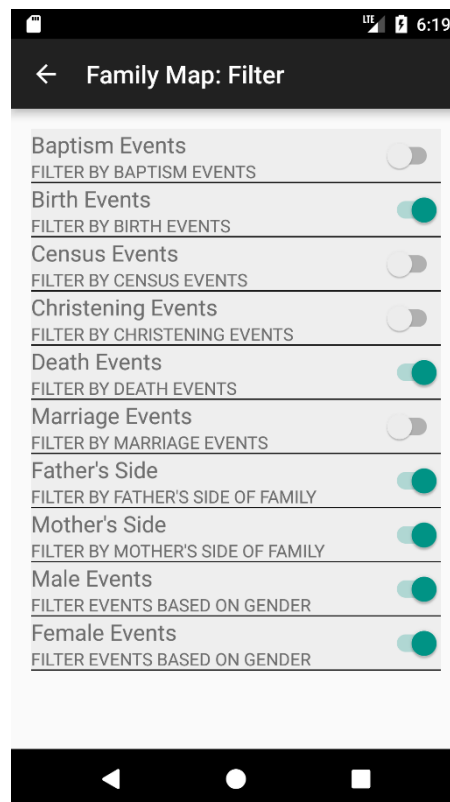
Lines are drawn connecting each event in a person's life story (i.e., the person associated with the selected event), ordered chronologically. (See the Person Activity section for information on how events are ordered chronologically.) If some of the person's life events are invisible due to the current event filter settings, connect whatever events are visible.

Up Button / Options Menu

The Settings Activity has an up button, which returns the user to the Main Activity. The Settings Activity has no options menu.

Filter Activity

The Filter Activity allows the user to select which filters are enabled and disabled, as shown below. Changing the filters changes which events are drawn on the map, shown in search results (search activity) and shown in the person's details (person activity). When a particular event type is turned off, events of that type should never appear on the map in MapFragment, in a search result in Search Activity, or as a life event in Person Activity. Filters control only which events are visible, and have no effect on which people are visible (e.g., in a search result).



Filter Types

The following filters are supported:

Filter by Event Type

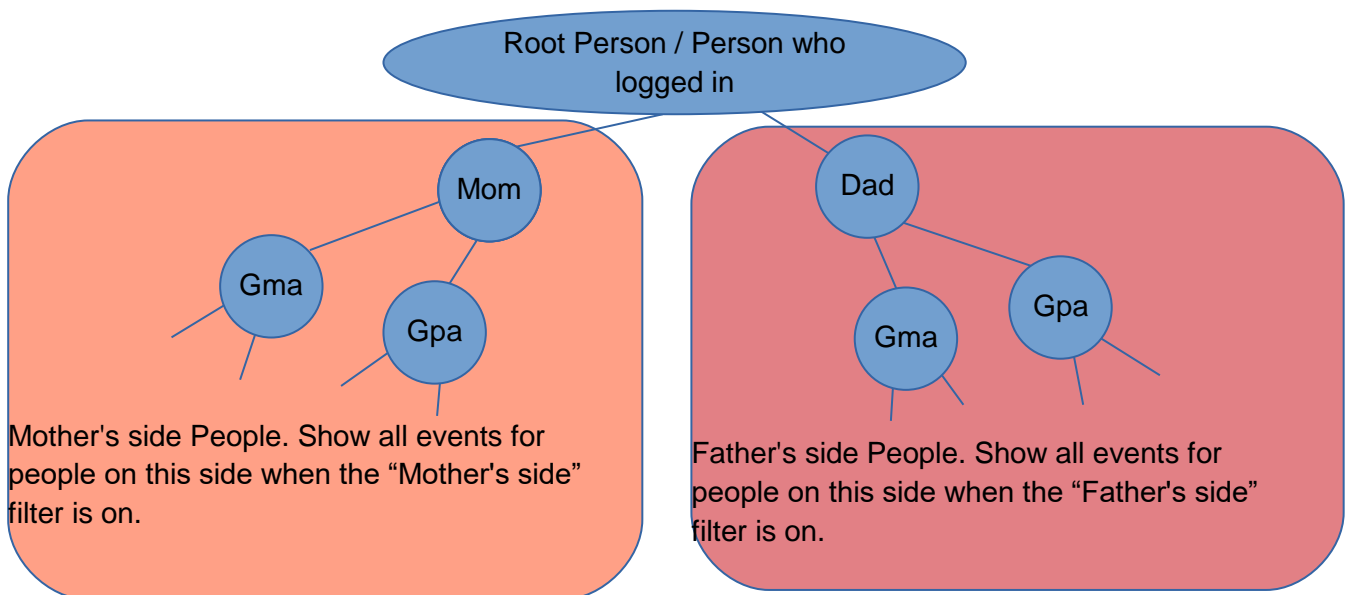
These filters allow the user to select one or more event types that will be displayed on the map. By default, all event types are selected. The event type list is built dynamically based on event descriptions that appear in the data. The list is case insensitive, so an event with the description "birth" is in the same type-group as an event with the description of "BIRTH". The map updates to only show events with the specified types.

Filter by Gender

These filters allow the user to filter events based on gender. The map updates to only show events associated with people of the selected genders. By default, the map shows events associated with both genders.

Filter by Side

These filters allow the user to select which sides of their family will be displayed (Mother's side and/or Father's side). To determine "Mother's side" and "Father's side" you start with the person who has logged in. That person is the root of the tree and all ancestors from that person's mother's side gets classified as "Mother's side". The same applies for the "Father's side". When someone logs in one of the items returned in the JSON is the person's id of the person who just logged in. This is the person id of the root of the tree. (See picture below). By default, the map shows both sides of the family.

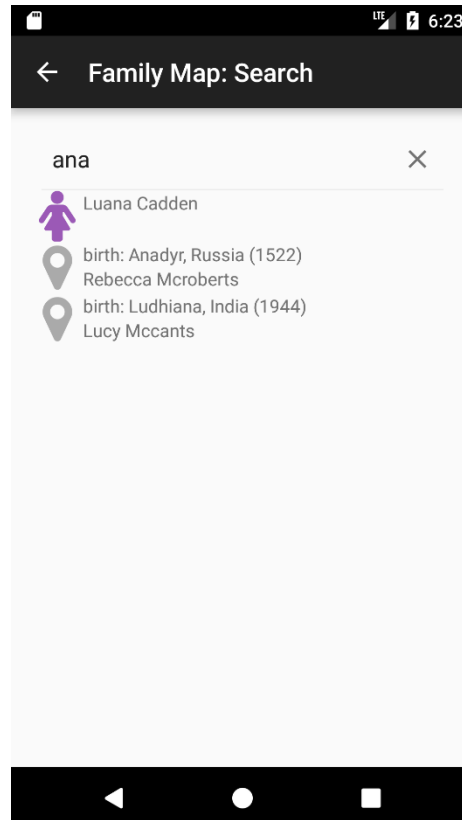


Up Button / Options Menu

The Filter Activity has an up button, which returns the user to the Main Activity. The Filter Activity has no options menu.

Search Activity

The Search Activity allows the user to search for people and events that match a specified search string. The user can enter a single search string. The specified string is searched for in people's first and last names, and in event's countries, cities, descriptions, and years. Case is ignored. All matching people and events (that aren't filtered) are listed in the search result, as shown below.



Layout

The Search Activity's layout displays a search input that accepts the user's search string. It also displays a list of search results, which is empty before a search has been executed. The results list displays a combination of both people and events, with the people items coming before event items. The people items display a person's name and gender. The event items display an event's type, city, country, date, and the associated person's name.

The Search Activity also has an up button, which returns the user to the Main Activity. It has no options menu.

Functionality

The search result list is empty until a search has been executed. When the user executes a search, the search result list is populated with all people and events that match the search string (and are not being filtered). The latest search result remains visible until another search is executed, or the activity is closed.

When the user selects a person from the search results list, a new Person Activity that contains the selected person's information is displayed. When the user selects an event from the search results list, a new Event Activity is displayed in which the event is automatically selected and the map is centered on the event.

The up button returns the user to the Main Activity.

Resources

The following resources will be helpful in developing your Family Map Android application.

Family Map Server

You will need to use the Family Map server that you created in the previous project.

IMPORTANT POINTS FOR CONNECTING YOUR CLIENT AND SERVER

- The Android device and the machine running the server should be on the same WiFi network. (We have seen in the Talmage building there tends to be an issue connecting while using BYU Secure and BYU Guest, but only in the Talmage Building. When in the Talmage building we recommend using CS Dept).
- Make sure to add the following permissions to the AndroidManifest.xml file to allow the Android device to access the network.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

These should be added just before the <application> entry in AndroidManifest.xml.

Map Libraries for Android

To help build the Map Fragment, you will use the Google Maps for Android library. This library provides an Android UI fragment that can display maps retrieved from Google Maps. The Google map fragment allows you to draw markers and lines on a map using latitude and longitude coordinates, and handles panning and zooming of maps, etc. Information on how to use the Google map fragment can be found here:

<https://developers.google.com/maps/documentation/android-api/start>

If you are developing for an Amazon Kindle device, instead of using the Google Maps for Android Library, you will use the Amazon Maps for Android (version 2) library. The Amazon Maps library is patterned after the Google Maps library, so the two are very similar. Information on how to use the Amazon Maps library can be found here:

<https://developer.amazon.com/public/apis/experience/maps/docs-v2/configuring-your-project-to-use-the-amazon-maps-api-v2>

NOTE: Google Play Services must be installed on an Android device in order to use the Google Maps library. Google Play Services is not installed on Amazon Kindle devices, and therefore you must use the Amazon Maps library on Kindle. However, with some effort it is possible to install Google Play Services on Kindle, thus allowing use of the Google Maps library on Kindle. If you want to do this, Google for details on how to do so.

Drawing Icons

Android Iconify is a library that makes it easy to create and draw dynamically-customizable icons in an Android application. This library is useful for drawing the following icons required by the Family Map application:

- Male and female gender icons
- Event icon
- Search, Filter, Settings, and Go to Top options menu icons
- Other icons you wish to incorporate in your UI

The web page for the Android Iconify library can be found here:

<https://github.com/JoanZapata/android-iconify>

To use Android Iconify, you should include a dependency similar to the following in the **build.gradle** file in your project's **app** folder:

```
dependencies {  
    ...  
    compile 'com.joanzapata.iconify:android-iconify:2.2.2'  
    compile 'com.joanzapata.iconify:android-iconify-fontawesome:2.2.2'  
    ...  
}
```

In your MainActivity's onCreate method, initialize the Iconify library as follows:

```
import com.joanzapata.iconify.Iconify;  
import com.joanzapata.iconify.fonts.FontAwesomeModule;
```



```
onCreate() {
    ...
    Iconify.with(new FontAwesomeModule());
    ...
}
```

You can easily create an Android drawable for an icon with code similar to the following:

```
import com.joanzapata.iconify.IconDrawable;
import com.joanzapata.iconify.fonts.FontAwsomelcons;
...
Drawable genderIcon = new IconDrawable(getActivity(), FontAwsomelcons.fa_male).
    colorRes(R.color.male_icon).sizeDp(40);
genderImageView.setImageDrawable(genderIcon);
```

The constant values for the necessary icons are:

```
fa_male (male icon)
fa_female (female icon)
fa_map_marker (event icon)
fa_search (search icon)
fa_filter (filter icon)
fa_gear (settings icon)
fa_angle_double_up ("go to top" icon)
```

JSON Parsing

You may use any library you wish for parsing and generating JSON data. If you choose to use the GSON library, you should include a dependency similar to the following in the **build.gradle** file in your project's **app** folder:

```
dependencies {
    ...
    compile 'com.google.code.gson:gson:2.8.0'
    ...
}
```

Automated Tests

Write JUnit tests to verify that your Server Proxy class works correctly. Specifically, you should test your Server Proxy methods for logging in, registering a new user, retrieving people, and retrieving events. For each method, you should test both success and failure cases.

Write JUnit tests to verify that your model is correctly:

- calculating family relationships (i.e., spouses, parents, children)
- filtering events according to the current filter settings
- chronologically sorting a person's individual events (birth first, death last, etc.)
- searching for people and events

Source Code/Test Case Evaluation

After you pass off your project with a TA, you should immediately submit your project source code for grading. Your grade on the project will be determined by the date you submitted your source code, not the date that you passed off. If we never receive your source code, you will not receive credit for the assignment. Here are the instructions for submitting your project source code:

1. In Android Studio, execute "Build -> Clean Project" (if this option is not available, you may skip this step).
2. Create a ZIP file containing ALL of your project's files (not just the Java files)
3. The name of the ZIP file should be your NetID. For example, if your NetID is "bob123", the name of your zip file should be "bob123.zip".
4. Submit your ZIP file through Learning Suite

To demonstrate that your test cases execute successfully, you should run your unit tests inside Android Studio, and make a screen shot of the successful test results displayed by Android Studio. Submit your screen shot through Learning Suite. This screen shot is submitted separately from your code ZIP file (i.e., they are different assignments in Learning Suite).

The following criteria will be used to evaluate your source code:

- (25%) Effective class, method, and variable names
- (30%) Effective decomposition of classes and methods
- (30%) Code layout is readable and consistent
- (15%) Effective organization of classes into Java packages

The TAs will also do a code review of your test cases in order to assess their quality.

Grading

Your grade on this project will consist of the following components:

1. Functionality (how well your app works)
2. Test Case Quality (quality of your test cases)

3. Source Code Quality (quality of your source code)

See the course policies page on the CS 240 web site for details on how much each of these components counts toward your overall course grade.